305-CD-005-001

EOSDIS Core System Project

Release A SDPS Client Subsystem Design Specification for the ECS Project

July 1995

Hughes Information Technology Corporation
Landover, MD

Release A SDPS Client Subsystem Design Specification for the ECS Project

July 1995

Prepared Under Contract NAS5-60000
CDRL Item # 046

APPROVED BY

Parag N. Ambardekar /s/ 7/25/95
Parag Ambardekar, Release A CCB Chairman Date
EOSDIS Core System Project

Hughes Information Technology Corporation
Landover, Maryland

This page intentionally left blank.

Preface

This document is one of sixteen comprising the detailed design specifications of the SDPS and CSMS subsystem for Release A of the ECS project. A complete list of the design specification documents is given below. Of particular interest are documents number 305-CD-004, which provides an overview of the subsystems and 305-CD-018, the Data Dictionary, for those reviewing the object models in detail. A Release A SDPS and CSMS CDR Review Guide (510-TP-002) is also available.

The SDPS and CSMS subsystem design specification documents for Release A of the ECS Project include:

305-CD-004	Release A Overview of the SDPS and CSMS Segment System Design Specification
305-CD-005	Release A SDPS Client Subsystem Design Specification
305-CD-006	Release A SDPS Interoperability Subsystem Design Specification
305-CD-007	Release A SDPS Data Management Subsystem Design Specification
305-CD-008	Release A SDPS Data Server Subsystem Design Specification
305-CD-009	Release A SDPS Ingest Subsystem Design Specification
305-CD-010	Release A SDPS Planning Subsystem Design Specification
305-CD-011	Release A SDPS Data Processing Subsystem Design Specification
305-CD-012	Release A CSMS Segment Communications Subsystem Design Specification
305-CD-013	Release A CSMS Segment Systems Management Subsystem Design Specification
305-CD-014	Release A GSFC Distributed Active Archive Center Implementation
305-CD-015	Release A LaRC Distributed Active Archive Center Implementation
305-CD-016	Release A MSFC Distributed Active Archive Center Implementation
305-CD-017	Release A EROS Data Center Distributed Active Archive Center Implementation
305-CD-018	Release A Data Dictionary for Subsystem Design Specification
305-CD-019	Release A System Monitoring and Coordination Center Implementation

Object models presented in this document have been exported directly from CASE tools and in some cases contain too much detail to be easily readable within hard copy page constraints. The reader is encouraged to view these drawings on line using the Portable Document Format (PDF) electronic copy available via the ECS Data Handling System (ECS) at URL http:// edhs1.gsfc.nasa.gov.

This document is a contract deliverable with an approval code 2. As such, it does not require formal Government approval, however, the Government reserves the right to request changes within 45 days of the initial submittal. Once approved, contractor changes to this document are handled in accordance with Class I and Class II change control requirements described in the EOS Configuration Management Plan, and changes to this document shall be made by document change notice (DCN) or by complete revision.

Data Management Office
The ECS Project Office
Hughes Information Technology Corporation
1616 McCormick Drive
Landover, MD 20785

Abstract

This document presents the design of the Data Management Subsystem of the Earth Observing System Data and Information System (EOSDIS) Core System (ECS). It defines the Data Management Subsystem's Release A CSCI and HWCI structures, as well as subsystem design based on Level 4 requirements.

Keywords: SDPS, data, management, CSCI, HWC/I, V0, gateway, ODL, client

This page intentionally left blank.

Change Information Page

List of Effective Pages	
Page Number	**Issue**
Title	Original
iii through x	Original
1-1 through 1-2	Original
2-1 through 2-4	Original
3-1 through 3-4	Original
4-1 through 4-12	Original
5-1 through 5-6	Original
AB-1 through AB-6	Original

Document History			
Document Number	**Status/Issue**	**Publication Date**	**CCR Number**
305-CD-005-001	Original	July 1995	95-0466

This page intentionally left blank.

Contents

5. WKBCH - Workbench CSCI

Figures

Tables

Abbreviations and Acronyms

1. Introduction

1.1 Identification

This Release A SDPS Client Subsystem Design Spefication for the ECS Project, Contract Data Requirement List (CDRL) Item 046, with requirements specified in Data Item Description (DID) 305/DV2, is a required deliverable under the Earth Observing System Data and Information System (EOSDIS) Core System (ECS), Contract NAS5-60000. This publication is part of a series of documents comprising the Science and Communications Development Office design specification for the Communications and System Management Segment (CSMS) and the Science and Data Processing Subsystem (SDPS) for Release-A.

1.2 Scope

The Release A SDPS Client Subsystem Design Spefication defines the progress of the design. It defines the Client Subsystem computer software and hardware architectural design, as well as subsystem design based on Level 4 requirements.

This subsystem is on an incremental development track. It is released in and reviewed in the form of Evaluation Packages (EP), and is, therefore, not part of the formal Release-A Critical Design Review. The overview material for these components has been included in this document for information purposes only.

1.3 Document Organization

The document is organized to describe the Client Subsystem design as follows:

Section 1	provides information regarding the identification, scope, status, and organization of this document.
Section 2	provides a listing of the related documents, which were used as source information for this document.
Section 3	provides an overview of the Subsystem, focusing on the high-level design concept. This provides general background information to put Client into context.
Sections 4 and 5	contain the structure of the computer software configuration items (CSCI) comprising the Client Subsystem.

The section Abbreviations and Acronyms contains an alphabetized list of the definitions for abbreviations and acronyms used in this document.

1.4 Status and Schedule

This submittal of DID 305/DV3 meets the milestone specified in the Contract Data Requirements List (CDRL) of NASA Contract NAS5-60000. The previous submittal was reviewed during the SDPS Preliminary Design Review (PDR) and this submittal reflects changes to the design which resulted from that review. The PDR also triggered a number of follow up actions in response to Review Item Discrepancies (RID) the results of which are incorporated into the Critical Design Review (CDR) version of this document.

2. Related Documents

2.1 Parent Documents

The parent document is the document from which the scope and content of this Release A SDPS Client Subsystem Design Spefication is derived.

194-207-SE1-001	System Design Specification for the ECS Project

2.2 Applicable Documents

The following documents are referenced within this Release A SDPS Client Subsystem Design Spefication, or are directly applicable, or contain policies or other directive matters that are binding upon the content of this document.

209-CD-011-001	Interface Control Document Between EOSDIS Core System (ECS) and the Version 0 System
308-CD-001-003	Software Development Plan for the ECS Project
423-41-03	Goddard Space Flight Center, EOSDIS Core System (ECS) Contract Data Requirements Document

2.3 Information Documents Not Referenced

The following documents, although not referenced herein and/or not directly applicable, do amplify and clarify the information presented in this document. These documents are not binding on the content of the SDPS Design Specifications.

205-CD-002-001	Science User's Guide and Operations Procedure Handbook for the ECS Project. Part 4: Software Developer's Guide to Preparation, Delivery, Integration, and Test with ECS
206-CD-001-002	Version 0 Analysis Report for the ECS Project
209-CD-010-001	Interface Control Document Between EOSDIS Core System (ECS) and the Langley Research Center (LaRC) Distributed Active Archive Center (DAAC) Draft
194-302-DV2-001	ECS Facilities Plan for the ECS Project
101-303-DV1-001	Individual Facility Requirements for the ECS Project, Preliminary
194-317-DV1-001	Prototyping and Studies Plan for the ECS Project
318-CD-003-XXX	Prototyping and Studies Progress Report for the ECS Project (monthly)
333-CD-002-003	SDP Toolkit Users Guide for the ECS Project
601-CD-001-002	Maintenance and Operations Management Plan for the ECS Project
604-CD-001-004	Operations Concept for the ECS Project: Part 1 -- ECS Overview
101-620-OP2-001	List of Recommended Maintenance Equipment for the ECS Project
194-703-PP1-001	System Design Review (SDR) Presentation Package for the ECS Project

193-801-SD4-001	PGS Toolkit Requirements Specification for the ECS Project
194-813-SI4-002	Planning and Scheduling Prototype Results Report for the ECS Project
194-813-SI4-003	DADS Prototype One FSMS Product Operational Evaluation
194-813-SI4-004	DADS Prototype One STK Wolfcreek 9360 Automated Cartridge System Hardware Characterization Report
813-RD-009-001	DADS Prototype Two Multi-FSMS Product Integration Evaluation
828-RD-001-002	Government Furnished Property for the ECS Project
193-WP-118-001	Algorithm Integration and Test Issues for the ECS Project
193-WP-611-001	Science-based System Architecture Drivers for the ECS Project, Revision 1.0
193-WP-623-001	ECS Evolutionary Development White Paper
194-WP-901-002	EOSDIS Core System Science Information Architecture, White Paper, Working Paper
194-WP-902-002	ECS Science Requirements Summary, White Paper, Working Paper
194-WP-904-002	Multi-Track Development for the ECS Project, White Paper, Working Paper
194-WP-913-003	User Environment Definition for the ECS Project, White Paper, Working Paper
194-WP-914-001	CORBA Object Request Broker Survey for the ECS Project, White Paper, Working Paper
194-WP-918-001	DADS Prototype One FSMS Product Operational Evaluation, White Paper, Draft Report
194-WP-925-001	Science Software Integration and Test, White Paper, Working Paper
420-WP-001-001	Maximizing the Use of COTS Software in the SDPS SDS Software Design, White Paper
193-TP-626-001	GCDIS/UserDIS Study ECS Technical Paper, Draft 0.2
194-TP-266-002	Data Distribution Architecture Logical Object Model (LOM) for the ECS Project, Version 2.01
194-TP-267-001	Data Server Architecture Logical Object Model (LOM) for the ECS Project, Version 2.00
194-TP-313-001	ECS User Characterization Methodology and Results
194-TP-316-002	Data Compression Study for the ECS Project
194-TP-548-001	User Scenario Functional Analysis [for the ECS Project]
194-TP-569-001	PDPS Prototyping at ECS Science and Technology Laboratory, Progress Report #4
222-TP-003-006	Release Plan Content Description for the ECS Project
430-TP-001-001	SDP Toolkit Implementation with Pathfinder SSM/I Precipitation Rate Algorithm, Technical Paper

440-TP-001-001	Science Data Server Architecture Study [for the ECS Project]
420-TD-001-001	ECS Data Server Taxonomy Technical Description
none	Hughes Training, Inc., ECS User Interface Style Guide, White Paper, Version 4.0
423-16-01	Goddard Space Flight Center, Data Production Software and Science Computing Facility (SCF) Standards and Guidelines
423-41-02	Goddard Space Flight Center, Functional and Performance Requirements Specification for the Earth Observing System Data and Information System (EOSDIS) Core System
540-022	Goddard Space Flight Center, Earth Observing System (EOS) Communications (Ecom) System Design Specification
560-EDOS-0211.0001	Goddard Space Flight Center, Interface Requirements Document Between EDOS and the EOS Ground System (EGS)

This page intentionally left blank.

3. Client Subsystem Overview

3.1 Subsystem Overview

The client subsystem provides a collection of components through which users access the services and data available in ECS and other systems interoperable with ECS. The client subsystem also includes the services needed to interface an application (e.g., a science algorithm) with ECS, e.g., for data access or to make use of ECS provided toolkits. Accessed services can be remote (i.e., via wide-area network to other sites) and local (e.g., to a database manager at the user's site).

Client subsystem components fall into one of the following general categories:

- Graphical and command line accessible application programs which implement the range of functionality available in the client.
- An intuitive desktop manager, which manages desktop objects (files) in the user's local file space, and which represent applications and data in the domain of the client.
- Version 0 Client as Release-A Client

In addition, the workstations which operate an ECS client subsystem will contain infrastructure support software which is part of CSMS and platform operating support software such as the vendor Operating System and its supporting software libraries.

3.2 Subsystem Structure

The Client subsystem for Release-A is composed of two CSCIs:

- Desktop CSCI (DSKT) is a software component. It provides a standardized GUI framework for the user's interaction with ECS software and data.
- Workbench CSCI (WKBCH) is a software component. It includes the initial set of ECS provided applications and libraries that provide access to ECS services. A major portion of Workbench functionality at Release-A is provided by the Version 0 Client application, a CSC within the Workbench at Release-A CSCI.

Figure 3.1-1 shows the subsystem context within ECS. For each of the context diagram flows, Table 3.1-1 contains a description of the subsytem interfaces.

Figure 3.1-1 Client Subsystem Context Diagram

Table 3-1-1. Subsystem Interfaces

Source	Destination	Data Types	Data Volume	Frequency
Client	Data Management	V0 ODL Inventory Search	low	as requested
Client	Data Management	V0 ODL Browse Request	low	as requested
Client	Data Management	V0 ODL Product Order Request	low	as requested
Client	Data Management	V0 ODL Directory Search	low	as requested
Client	Interoperability	Search requests	low	as requested
Client	Interoperability	Subscriptions	low	as requested
Client	Interoperability	Advertisements, Software and Documents	low	as supplied by user
Client	MSS	User Registration Status Requests	low	as required
Client	MSS	User Registration Requests	low	as required
Data Management	Client	Results Set	low-high	in response to request
Data Management	Client	Granules Metadata	low	in response to request
Data Management	Client	Error and status info	low	as available or requested
Interoperability	Client	Advertisements	low	in response to request
Interoperability	Client	Notifications	low	in response to subscription
Interoperability	Client	Search results	low-medium	in response to request
MSS	Client	User registration status	low	as required

In the table, where an exact number is unavailable, the data volume is estimated as low (less than 1 MB), medium (between 1 MB and 1 GB), or high (greater than 1 GB) per use defined in the frequency column. The frequency information will be updated as the interfaces are fully defined.

3.3 Subsystem Design Rationale

The client design drivers are a set of requirements for creating a software system which is supported on HP, Sun, SGI, DEC, and IBM platforms and which users can either install on their local workstation or access remotely. Required capabilities include obtaining data and services from ECS and non-ECS entities external to the client, as well as submitting data, software, and documentation into the ECS. To meet the challenge of creating a friendly, easy to use interface to the broad range of ECS capabilities, the client desktop paradigm was developed. This consists of a desktop manager which manages a set of desktop objects, which are represented iconically. Desktop objects represent files on the user's workstation. The set of core client application programs is collectively called "the Workbench". The need to manage the desktop and objects under its control was the main driver for the desktop's design. Further, the specific requirements for the client to send and receive ECS and non-ECS data, software, and advertisements, access services, provide user feedback, create hypertext-based documents, track client activities, visualize ECS data, obtain definitions of terms and acronyms, and register as an ECS user, drove the design of the Workbench applications.

The elapsed time available for the Release-A development has been progressively shortened since the original ECS development schedule was set. This has resulted in the prototyping/incremental development schedule being tightened such that the feedback from tire kickers and the user community into succeeding prototypes and increments will be severely compressed and as a result potentially have less immediate impact on the development than is desirable.

In order to mitigate this situation, the system level IMS components from V0 (which is also termed as System-V0) will be used to substitute for the functionality that the Client and Data Management Subsystems would provide in V1, in the 9 month period until full V1 functionality is released for these subsystems at Release B. This approach will permit sufficient time for user feedback to be incorporated into the V1 Client and Data Management components before they are incorporated into Release B. The system level components are those parts of V0 system which provide a unified view across all of the V0 DAACs. This includes the GUI/ChUI client applications, search agent (IMS client) and local server mapping components. The System-V0 will be incrementally enhanced so that it can support access to the V1 components at Release-A, always provided that these enhancements do not step outside of the overall V0 mission and that the implementation of these enhancements do not negatively impact the implementation of DAAC requested enhancements.

It is important to stress that there is no intention to incrementally develop the reused System-V0 components at each release beyond Release-A. The reuse described here is only to address the early schedule and user migration issues at Release-A; some important conceptual changes need to be introduced at Release B with full V1 functionality to make best use of the developing technology and underlying provider level capabilities that will be available across all DAACs at Release B. For example,

- the underlying client-server interaction mechanisms
- the user session concept
- integrated schema management across all services.

4. DESKT - Desktop CSCI

4.1 CSCI Overview

The Desktop CSCI provides a desktop environment, which consists of a set of user and client files (called desktop objects, each with a representative icon), a pop-up menu for quick access to specific client applications, specific operations defined for desktop objects, including drag and drop, double-click to open, copy, delete, and rename, and a *desktop manager* , which manages the set of desktop objects. Desktop objects include, but are not limited to: search parameters, result-sets, documents, subscriptions, request-sets, data granules, project folders, application programs, and configuration files. Managing the desktop objects includes viewing them in a hierarchical format, viewing their attributes, e.g. name, creation date, and status. The status is applicable only to certain classes of objects, such as product or search requests, and provides a convenient means of providing information about asynchronous requests from the user.

Note that the Desktop Manager will contain context-sensitive help.

The desktop is either represented in hierarchical format or iconic format. The desktop consists of desktop objects. There are three types of desktop objects; Container objects, Application objects and Document objects. Each desktop object will have zero or more actions attached to it. When a desktop object is invoked by double-clicking on it, it invokes the default action associated with it. Each desktop object is represented with a specific icon on the desktop.

(See Attached OMT Diagram.)

4.2 CSCI Context

The Desktop provides no public interfaces for access by the formal track CSCI.

4.3 CSCI Object Model

This section provides the object model for the Desktop CSCI. The model is graphically depicted in Figure 4.3-1. Each of the objects in this figure are described in detail in the subsequent paragraphs.

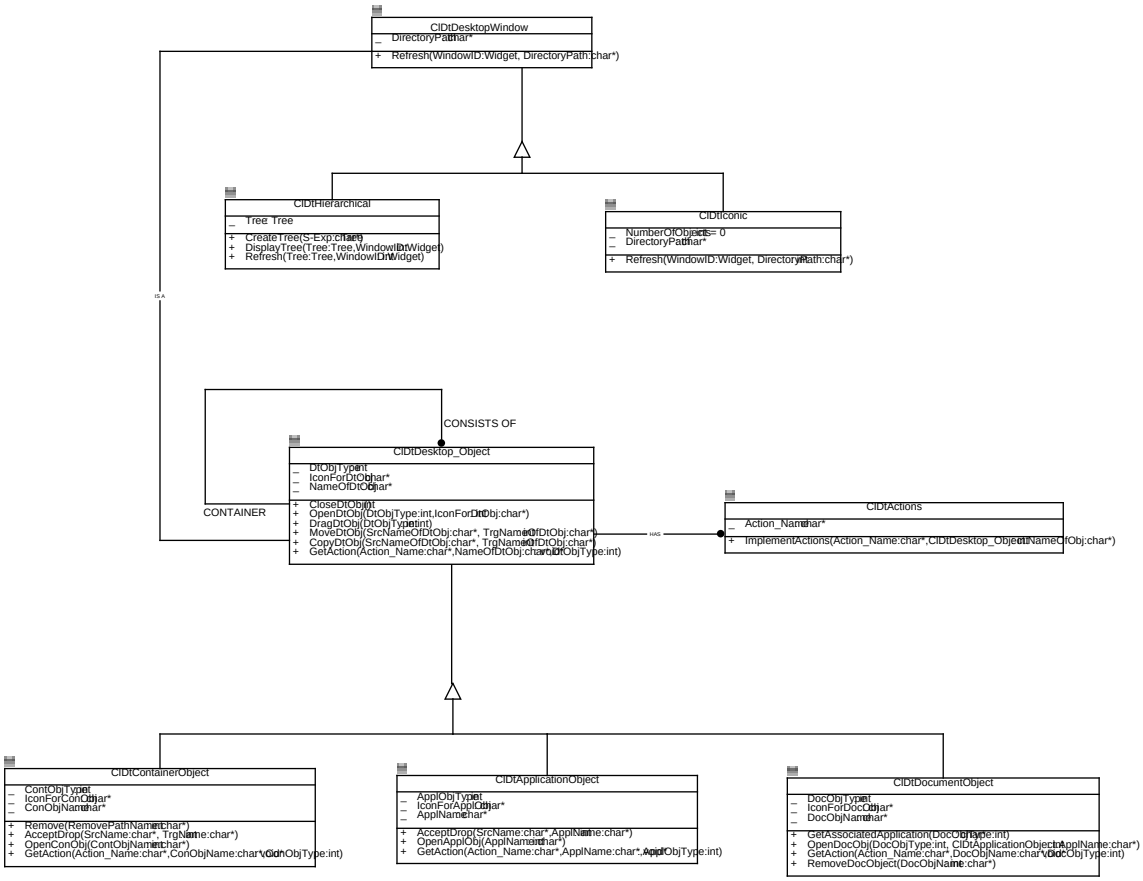

Figure 4.3-1. Object Model Diagram

4.3.1 ClDtActions Class

Parent Class: Not Applicable
Public: No Distributed Object: No
Purpose and Description:

An Action is an Event to be performed by the Desktop on user's or system's request. For example; OpenApplication, CloseApplication, DragIcon, CopyObject, OpenNewWindow and RefreshScreen etc. This Class contains implementations of all actions supported by Desktop objects. The set of Actions to be supported by a Desktop object will be based on its type.

Attributes:

Action_Name - Name of the action
Data Type: char*
Privilege: Private
Default Value:

Operations:

ImplementActions - Implementation of the given action
Arguments: Action_Name:char*,ClDtDesktop_Object.NameOfObj:char*
Return Type: int
Privilege: Public

Associations:

The ClDtActions class has associations with the following classes:
Class: ClDtDesktop_Object HAS

4.3.2 ClDtApplicationObject Class

Parent Class: ClDtDesktop_Object
Public: No Distributed Object: No
Purpose and Description:
Application object is a type of desktop object. It provides default behaviour for objects which represent executable programs.

Attributes:

ApplName - Name of Application Object
Data Type: char*
Privilege: Private
Default Value:

ApplObjType - Type of Application Object
Data Type: int
Privilege: Private
Default Value:

IconForApplObj - Filename of the pixmap to be used as icon for Application object
Data Type: char*
Privilege: Private
Default Value:

Operations:

AcceptDrop - Accepts the related data files as drop input and opens the application.
Arguments: SrcName:char*,ApplName:char*
Return Type: int
Privilege: Public

GetAction - Get the function which implements the specified action for the application object
Arguments: Action_Name:char*,ApplName:char*,ApplObjType:int
Return Type: void*
Privilege: Public

OpenApplObj - Open the Application
Arguments: ApplName:char*
Return Type: int
Privilege: Public

Associations:

The ClDtApplicationObject class has associations with the following classes:
None

4.3.3 ClDtContainerObject Class

Parent Class: ClDtDesktop_Object
Public: NoDistributed Object: No
Purpose and Description:

A type of Desktop Object which is capable of holding other desktop objects (containers, applications, and documents). Since containers can contain other container objects, they can be used to implement hierarchical structures. Actions associated with this object are display contents, move into container, remove from container etc.

Attributes:

ConObjName - Name of the container object
Data Type: char*
Privilege: Private
Default Value:

ContObjType - Type of container objects (for ex. directory)
Data Type: int
Privilege: Private
Default Value:

IconForConObj - Filename of the pixmap to be used as icon for the container object
Data Type: char*
Privilege: Private
Default Value:

Operations:

AcceptDrop - A container object accepts a drop of any other desktop objects.
Arguments: SrcName:char*, TrgName:char*
Return Type: int
Privilege: Public

GetAction - Get the function which implements the specified action for container objects
Arguments: Action_Name:char*,ConObjName:char*,ConObjType:int
Return Type: void*
Privilege: Public

OpenConObj - Open the container object to show all the desktop objects contained in it.
Arguments: ContObjName:char*
Return Type: int
Privilege: Public

Remove - Removes the container object if it does not contain any other desktop objects.
Arguments: RemovePathName:char*
Return Type: int
Privilege: Public

Associations:

The ClDtContainerObject class has associations with the following classes:
None

4.3.4 ClDtDesktopWindow Class

Parent Class: Not Applicable
Public: No Distributed Object: No
Purpose and Description:
The Desktop Window can be either in hierarchical format or iconic format. Desktop Window contains zero or more desktop objects. Desktop Window itself is a Desktop Object i.e., a container object.

Attributes:

DirectoryPath - Gives the Directory path of the Desktop Window
Data Type: char*
Privilege: Private
Default Value:

Operations:

Refresh - Refresh the Desktop Window with the contents of the DirectoryPath in the specified Widget
Arguments: WindowID:Widget, DirectoryPath:char*
Return Type: Void
Privilege: Public

Associations:

The ClDtDesktopWindow class has associations with the following classes:
Class: ClDtDesktop_Object ISA

4.3.5 ClDtDesktop_Object Class

Parent Class: Not Applicable
Public: No Distributed Object: No
Purpose and Description:
Desktop object can be a container object, Application object or a document object. Each object will have zero or more related actions. Each desktop object will have an icon associated with it. A desktop object will be displayed using this icon.

Attributes:

DtObjType - Type of Desktop Object...
Data Type: int
Privilege: Private
Default Value:

IconForDtObj - Filename of the pixmap to be used as icon for Desktop Object
Data Type: char*
Privilege: Private
Default Value:

NameOfDtObj - Desktop Object Name
Data Type: char*
Privilege: Private
Default Value:

Operations:

CloseDtObj - Close the Desktop object
Arguments:
Return Type: int
Privilege: Public

CopyDtObj - Copy the Source Desktop Object to the specified target
Arguments: SrcNameOfDtObj:char*, TrgNameOfDtObj:char*
Return Type: int
Privilege: Public

DragDtObj
Arguments: DtObjType:int
Return Type: int
Privilege: Public

GetAction - Get an action(s) from ClDtActions objects which is related to the desktop object of given type.
Arguments: Action_Name:char*,NameOfDtObj:char*,DtObjType:int
Return Type: void*
Privilege: Public

MoveDtObj - Move the Source Desktop Object to the specified target
Arguments: SrcNameOfDtObj:char*, TrgNameOfDtObj:char*
Return Type: int
Privilege: Public

OpenDtObj - Open the Desktop Object of given type
Arguments: DtObjType:int,IconForDtObj:char*
Return Type: int
Privilege: Public

Associations:

The ClDtDesktop_Object class has associations with the following classes:
Class: ClDtActions HAS
Class: ClDtDesktopWindow ISA

4.3.6 ClDtDocumentObject Class

Parent Class: ClDtDesktop_Object
Public: No Distributed Object: No
Purpose and Description:
A type of Desktop object which provides a mechanism for associating multiple types of descriptive data with an object. Applications can be associated with a class of desktop document objects, and this application can be invoked to access the data described in the document object.

Attributes:

DocObjName - Name of Document Object
Data Type: char*
Privilege: Private
Default Value:

DocObjType - Type of Document Object
Data Type: int
Privilege: Private
Default Value:

IconForDocObj - Filename of the pixmap to be used as icon for Document Object
Data Type: char*
Privilege: Private
Default Value:

Operations:

GetAction - Get the function which will be used to implement the specified action of the Document Object
Arguments: Action_Name:char*,DocObjName:char*,DocObjType:int
Return Type: void*
Privilege: Public

GetAssociatedApplication - Every document object will have a related application for manupulating the data. This operation will be used to get the related application
Arguments: DocObjType:int
Return Type: char*
Privilege: Public

OpenDocObj - Open the Document object using the related application
Arguments: DocObjType:int, ClDtApplicationObject.ApplName:char*
Return Type: int
Privilege: Public

RemoveDocObject - Document objects can be removed. Hence this operation will be used to remove the specified document object.
Arguments: DocObjName:char*
Return Type: int
Privilege: Public

Associations:

The ClDtDocumentObject class has associations with the following classes:
None

4.3.7 ClDtHierarchical Class

Parent Class: ClDtDesktopWindow
Public: No Distributed Object: No
Purpose and Description:
Displays desktop objects in hierarchical format.

Attributes:

Tree - A datastructure to represent the directory structure in a tree form.
Data Type: Tree
Privilege: Private
Default Value:

Operations:

CreateTree - Creates and returns a Tree from a S-Exp.
Arguments: S-Exp:char*
Return Type: Tree
Privilege: Public

DisplayTree - Displays a Tree(Directory structure) on the specified window.
Arguments: Tree:Tree,WindowID:Widget
Return Type: int
Privilege: Public

Refresh - Refresh by displaying the specified Tree in the Window
Arguments: Tree:Tree,WindowID:Widget
Return Type: int
Privilege: Public

Associations:

The ClDtHierarchical class has associations with the following classes:
None

4.3.8 ClDtIconic Class

Parent Class: ClDtDesktopWindow
Public: No Distributed Object: No
Purpose and Description:

Displays Desktop objects in Iconic form.

Attributes:

DirectoryPath - Directory Path to be displayed in the desktop.
Data Type: char*
Privilege: Private
Default Value:

NumberOfObjects - Number of DesktopObjects in the Iconic Display
Data Type: int
Privilege: Private
Default Value: 0

Operations:

Refresh - Refresh by displaying the contents of the specified directory.
Arguments: WindowID:Widget, DirectoryPath:char*
Return Type: int
Privilege: Public

Associations:

The ClDtIconic class has associations with the following classes:
None

4.4 CSCI Structure

The following table provides a summary of the components which make up this CSCI.

Table 4.6-1. DESKT Components

Name	Description	DEV or OTS	Release
Desktop	To manage a set of Desktop Objects	DEV	A

4.5 Desktop Dynamic Model

Since the Client subsystem is on an incremental development track and is reviewed in the form of Evaluation (EP), this section will be available after EP6 evaluation is completed.

4.6 CSCI Structure

Since the Client subsystem is on an incremental development track and is reviewed in the form of Evaluation (EP), this section will be available after EP6 evaluation is completed.

4.7 CSCI Management and Operation

Since the Client subsystem is on an incremental development track and is reviewed in the form of Evaluation (EP), this section will be available after EP6 evaluation is completed.

This page intentionally left blank.

5. WKBCH - Workbench CSCI

5.1 CSCI Overview

The Workbench is a set of application programs which implement the core functionality of the client. This includes requirements for locating and retrieving advertisements, data holdings, services, and a variety of documents. It also includes the capability to visualize data, either prior to requesting it (via browse image) or after it is delivered to the user's desktop. Other capabilities include the ability to send desktop objects via e-mail, to log and review transactions between the client and the external environment, to easily create hypertext documents, to provide comments and user feedback, to read and post news articles on an ECS bulletin board, to request definitions of ECS terms and acronyms, to configure the desktop environment and application-specific defaults, and to register for privileges beyond those of a guest.

5.2 CSCI Context

The context diagram for the Workbench CSCI is identical to that of the Client subsystem, since the Desktop CSCI has no external interfaces.

5.3 CSCI Structure

The following table provides a summary of the components which make up this CSCI, to the extent they are currently known. Since this CSCI is developed incrementally, the table presents our current best estimate of the CSCI components, but is likely to change as the CSCI evolves.

Table 5.3-1. WKBCH Components

Name	Description	DEV or OTS	Release
Release-A Client	Reuse of V0 Client	OTS	A
Hypertext Viewer CSC	HTML Viewer Application	OTS	A
Data Visualization (EOSView) CSC	EOS-HDF Data Viewer and Analysis Application	DEV	A
SDPS Toolkit CSC	Tools for accessing ECS services via APIs.	DEV	A
CSMS Toolkit CSC	ECS Systems Applications.	DEV	A

5.3.1 Release-A Client CSC

The System-V0 client is retained for Release-A as the primary user search agent, receiving inputs from users via the System-V0 ChUI and GUI interfaces. (*refer to description in Section 5.3.1.2)*

5.3.1.1 Purpose

The EOSDIS V0 IMS is a cooperative effort between various archives around the country and the Goddard Space Flight Center (GSFC) Earth Science Data Information System (ESDIS) project. This system allows users to search for and order data from several data centers in a single session. The goals of the EOSDIS Release-A Client are to facilitate earth science research through improved access to existing data and to serve as a testbed for the EOSDIS Core System (ECS).

5.3.1.2 Description

The V0 IMS consists of a Graphical User Interface (GUI) and a Character User Interface (ChUI). The Graphical User Interface (GUI) is a graphical environment which operates under the X-Window system, allowing the user to display multiple windows simultaneously, and supports a mouse for easy user interaction. It also allows search areas to be specified from a global map, and provides an interactive data browse facility and coverage map of data products. This interface requires Internet protocol support to function properly.

The IMS V0 Client provides the following services through its easy to use interface:

- Directory Information for Data Sets - provides brief concise high-level information about datasets from any point in the system.
- Guide Subsystem - provides detailed descriptions about datasets, platforms, sensors, projects, and data centers.
- Inventory - provides descriptions of specific observations or collections of observations of data (granules) that are available for request from a data archive.
- Coverage Maps - are two-dimensional graphical representation of the geographic coverage of selected inventory granules. It displays the Earth in an orthographic projection.
- Browse - allows a user to locate and retrieve reduced resolution images as an aid to data selection. The user may either view the image in the IMS interface or have it staged for FTP pickup.
- Product Request - allows users to view information pertaining to orderable data products, and then construct a request which is forwarded to the relevant archive for order processing.
- Access to the Global Change Master Directory - a multidisciplinary database of information about Earth and space science data. It contains high level descriptions of dataset holdings of various agencies and institutions.

The Character User Interface (ChUI) is intended for users who do not have access to an X-terminal, have a small monitor screen, or are accessing the system via low-bandwidth communications (e.g. via modem). Although it is designed to run on a VT100-standard terminal, it also operates on VT2xx- and VT3xx-class terminals, as well as other systems (e.g. personal computers with telecommunications software) that support VT100 terminal emulation.

5.3.2 Hypertext Viewer CSC

5.3.2.1 Purpose

The Hypertext Viewer CSC provides access to ECS provided services via HTTP/HTML protocols. Examples of such services include:

- Advertising Client Interface
- Context Sensitive Help Interface
- User Profile Interface
- User Registration Interface

Due to the explosive growth of hypertext technology, a trade study was performed to determine the hypertext viewer requirements for the Client subsystem .The purposes of the Hypertext Viewing Tool trade study were:

- to find a hypertext viewer for use as part of the Client Workbench for displaying HTML documents;
- to provide general access for Client users to the World Wide Web; and
- to serve as the implementation mechanism for the Client's hypertext based user interface.

5.3.2.2 Description

The Hypertext Document Viewing Tool trade study was defined in DID 211, Trade-off Studies Analysis Data for the ECS project. The trade study was accomplished by evaluating the currently (Spring, 1995) available Graphical Web browsers running on UNIX/X Windows. It resulted in the selection of Netscape. The following is a brief summary of the study:

This trade study was a joint project by ECS and University of Maryland at College Park (UMCP). UMCP provided technical expertise on OTSO, a systematic process for reusable software component selection; ECS performed searching of tools, and conducted hands-on evaluation. Criteria definition and results analyses were performed by both parties.

A total of over 30 tools were found during the search for possible tools. The search was carried out using the World Wide Web (WWW), as it was assumed such tools would most likely be found on the Web.

The tools were screened based on the following criteria:

- HTML level 3 compatibility. (the tool should support HTML level 3);
- Availability on UNIX, Mac, and MS-Windows platforms;
- Popularity of the tool (the tool should be one of the most widely used tools); and
- Availability (a working version of the tool must be available).

The tools which passed the initial screening were evaluated in more detail. This was based on a set of hierarchical criteria. The evaluation phase included generating explicit and detailed definitions for these criteria. The actual evaluation consisted of a set of tests corresponding to the criteria. The rigorous criteria definitions allowed a consistent evaluation of the tools even though several evaluators were involved. Each tool was evaluated by two evaluators, and each evaluator wrote a report describing the performance of each tool in each of the tests. The rationale for redundancy in evaluation was to improve evaluation consistency.

Evaluation results were discussed in a meeting where all but one evaluator was present. Most of the conflicting observations and open issues were resolved during the meeting. All remaining issues were resolved through assigned action items. Definition for two of the evaluation criteria tests were changed, and some of the tests were dropped due to the unavailability of appropriate data. The evaluators also found that some of the evaluation test definitions were too general and were thus not well understood.

Finally the evaluation results were analysed using two techniques: a commonly used weighted scoring technique, and a technique called Analytic Hierarchy Process (AHP). This two-fold analysis was performed to see whether the choice of scoring technique influenced the result. Although there were significant differences between the two methods, the outcome for best tool was the same in both techniques, namely Netscape.

5.3.3 Data Visualization (EOSView) CSC

EOSView is a tool for examining and verifying HDF-EOS data files. It is designed to be used in stages by an EOS user. The first stage is to use visualization to help in the selection of data. The user next needs to make sure that the received data is what is desired and visualization is used for the verification of data. Finally analysis is performed on the data, and results of the analysis presented.

EOSView provides the following groups of capabilities:

1. Selection of Data- The EOS client will use EOSView to visualize browse images during the selection of data. The EOS client and EOSView will execute simultaneously on the user workstation, and will communicate with each other via scripts.
2. Verification of Data- For verification of data, EOSView is designed to work on all major platforms and with all types of EOS data (i.e. data written in HDF- EOS).
3. Presentation and Analysis of Data- EOSView will be used to get EOS data in leading visualization and analysis systems. EOSView will provide file conversion routines in the program itself. And since EOSView can be driven by external scripts, these conversions can be initiated directly from the visualization systems (such as interface definition language (IDL)).

Documentation for EOSView can be found in: HDF-EOS Primer for Version 1 EOSDIS (175-WP-001-001) in Section 5.3.3.1 Purpose and Description.

5.3.4 SDPS Toolkit CSC

The Science Data Processing (SDP) Toolkit will be used by instrument science team developers in their Science Computing Facilities (SCFs) to develop EOS data production software and to prepare that software for integration into Distributed Active Archive Centers (DAACs). Subsequent usage of the Toolkit will be in conjunction with the services provided by the DAACs to produce, archive and distribute standard products.

Tools included are in the following groupings of functionality: process control, error and status message handling, ancillary data access, metadata creation and access, geo-coordinate transformations, geolocation, constants and unit conversion, time transformation, graphics and statistic libraries, Level 0 data access, platform ephemeris and attitude data access, celestial body position and general file I/O. The DAAC version of the SDP toolkit will have the same user API

as the SCF version. Additionally the DAAC version will contain an MSS event handling interface. The DAAC toolkit will be driven by a process control file rather than a manually created file in SCF toolkit operation.

The Toolkit is described in detail in the SDP Toolkit Users Guide for the ECS Project (333-CD-001-002) in Section 5.3.4.1 Purpose and Description

5.3.5 CSMS Toolkit CSC

The CSMS Toolkit is represented by a set of software that is made available to ECS users. It is a set of software or forms that enables users to gain access to ECS services. The toolkit is a listing or posting on the ECS Bulletin Board. The posting provides pointers to the files where the software is maintained. The posting and the software is maintained under software configuration control. The posting also contains information on the supported platforms and their operating systems (different files or ports of each software may be required). Users access and retrieve the referenced files via ftp.

The software consists such software items as Trouble Ticketing Package and HTML forms for entering and displaying status of trouble tickets; User registration forms; Kerberos client (for secure communications from non-DCE platforms); Kerberized versions of telnet (Ktelnet) and ftp (Kftp); ECS supported news readers and WEB clients; and checkpointed ftp (Cfpt).

This page intentionally left blank.

Abbreviations and Acronyms

ACMHW	Access Control and Management HWCI
ADC	Affiliated Data Center
ADS	Archive data sets
ADSHW	Advertising Service HWCI
ADSRV	Advertising Service CSCI
AITHW	Algorithm Integration & Test HWCI
AITTL	Algorithm Integration and Test Tools (CSCI)
AM	Ante meridian
ANSI	American National Standards Institute
APC	Access/Process Coordinators
API	Application Programming Interface
APID	Application Process Identifier
AQAHW	Algorithm QA HWCI
ASAP	As soon as possible
ASCII	American Standard Code for Information Interchange
ASF	Alaska SAR Facility (DAAC)
ATM	Asynchronous Transfer Mode
CD ROM	Compact disk read only memory
CDRL	Contract Data Requirements List
CERES	Clouds and Earth's Radiant Energy System
CI	Configuration Item
CIESIN	Consortium for International Earth Science Information Network
CLS	Client Subsystem
COTS	Commercial off-the-shelf
CPU	Central processing unit
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CCSDS	Consultative Committee for Space Data Systems
CM	Configuration Management
CSDT	Computer Science Data Types
CSMS	Communications and Systems Management Segment
CSS	Communication Subsystem (CSMS)

DAA	DAN Acknowledge
DAAC	Distributed Active Archive Center
DADS	Data Archive and Distribution System
DAN	Data Availability Notice
DAO	Data Assimilation Office
DAR	Data Acquisition Request
DAS	Data Availability Schedule
DBA	Database administrator
DBMS	Database Management System
DDA	Data Delivery Acknowledgement
DDICT	Data Dictionary CSCI
DDIST	Data Distribution CSCI
DDN	Data Delivery Notice
DDSRV	Document Data Server CSCI
DESKT	Desktop CI
DEV	Developed code
DID	Data Item Description
DIM	Distributed Information Manager
DIMGR	Distributed Information Management CSCI
DIPHW	Distribution & Ingest Peripheral Management HWCI
DMGHW	Data Management HWCI
DMS	Data Management System
DMS	Data Management Subsystem
DP	Data Processing
DPR	December Progress Review
DPREP	Science Data Pre-Processing CSCI
DPS	Data Processing Subsystem
DR	Data Repository
DRPHW	Data Repository HWCI
DS	Data Server
DSM	Distribution Storage Management
DSS	Data Server Subsystem
DT	Data Type
ECS	EOSDIS Core System
EDC	EROS Data Center (DAAC)
EDOS	EOS Data and Operations System

EOC	Earth Observation Center (Japan)
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information System
EP	Evaluation Package
EP	Early Prototype
ESDIS	Earth Science Data and Information System
ESDT	Earth Science Data Types
F&PRS	Functional and Performance Requirements Specification
FC	Fiber Channel
FDDI	Fiber distributed data interface
FDF	Flight Dynamics Facility
FOS	Flight Operations Segment
FSMS	File and Storage Management System
Ftp	File transfer protocol
GB	Gigabyte
GDAO	GSFC Data Assimilation Office
GFLOPS	Giga (billions) Floating Point Operations per Second
GOES	Geostationary Operational Environmental Satellite
GRIB	Gridded Binary
GSFC	Goddard Space Flight Center
GTWAY	Version 0 Interoperability Gateway CSCI
GUI	Graphic user interface
HDF	Hierarchical Data Format
HiPPI	High Performance Parallel Interface
HMI	Human machine interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
HWCI	Hardware Configuration Item
I&T	Integration and Test
I/O	Input/Output
ICD	Interface Control Document
ICLHW	Ingest Client HWCI
IDL	Interface Definition Language
IEEE	Institute of Electrical and Electronics Engineers
IERS	International Earth Rotation Service
IMS	Information Management Subsystem

IP	International Partner
IR-1	Interim Release 1
IRD	Interface Requirements Document
IS	Ingest Subsystem
ISS	Internetworking Subsystem (CSMS)
JPL	Jet Propulsion Laboratories
LaRC	Langley Research Center
LIM	Local Information Manager
LIMGR	Local Information Management CSCI
LIS	Lightning Imaging Sensor
L0	Level 0
MB	Megabyte
Mbps	Megabits per second
MBps	Megabytes per second
MD	Maryland
MFLOP	Millions of Floating Point Operations per Second
MOC	Mission Operations Center
MODIS	Moderate-Resolution Imaging Spectrometer
MPP	Massively Parallel Processor
MRF	Medium Range Forecast
MSFC	Marshall Space Flight Center
MSS	Management Subsystem (CSMS)
MTBF	Mean time between failures
MTTR	Mean time to restore
NESDIS	National Environmental Satellite Data and Information Service
NMC	National Meteorological Center
NOAA	National Oceanic and Atmospheric Administration
NSIDC	National Snow and Ice Data Center (DAAC)
O/A	Orbit/Attitude
ODC	Other Data Center
ODL	Object Description Language
ORNL	Oak Ridge National Laboratory (DAAC)
OSM	Open Storage Manager
OTS	Off-the-shelf
PAM	Permanent Archive Manager
PCI	Periphewral Component Interface

PDPS	Planning and Data Processing System
PDR	Preliminary Design Review
PDS	Production Data Set
PDS	Production Data Specialist
PGE	Product Generation Executive
PGS	Product Generation System
PLNHW	Planning HWCI
POSIX	Portable Operating System for UNIX
PRONG	Processing CSCI
Q	Quarter
Q/A	Quality Assurance
QA	Quality Assurance
QAC	Quality and Accounting Capsule
RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
REL	Release
RID	Review Item Discrepancy
RMA	Reliability, Maintainability, Availability
RTF	Rich Text Format
S/C	Spacecraft
SAA	Satellite Active Archives (NOAA)
SCF	Science Computing Facility
SCSI II	Small Computer System Interface
SDF	Software Development File
SDP	Science Data Processing
SDPF	Sensor Data Processing Facility (GSFC)
SDPS	Science Data Processing Segment
SDPS/W	Science Data Processing Software
SDPTK	SDP Toolkit CSCI
SDSRV	Science Data Server CSCI
SFDU	Standard Format Data Unit
SMC	System Management Center
SMP	Symmetric Multi-Processor
SPRHW	Science Processing HWCI
STMGT	Storage Management CSCI
TBD	To be determined

TBR	To be resolved
TDRSS	Tracking and Data Relay Satellite System
TONS	TDRSS Onboard Navigation System
TRMM	Tropical Rainfall Measuring Mission
TSDIS	TRMM Science Data and Information System
UR	Universal Reference
USNO	United States Naval Observatory
V0	Version 0
VC	Virtual Channel
VCDU-ID	Virtual Channel ID
WAIS	Wide Area Information Servers
WAN	Wide Area Network
WKBCH	Workbench CI
WKSHC	Working Storage HWCI
W/S	Workstation
WORM	Write Once Read Many
WS	Working Storage
WWW	World Wide Web